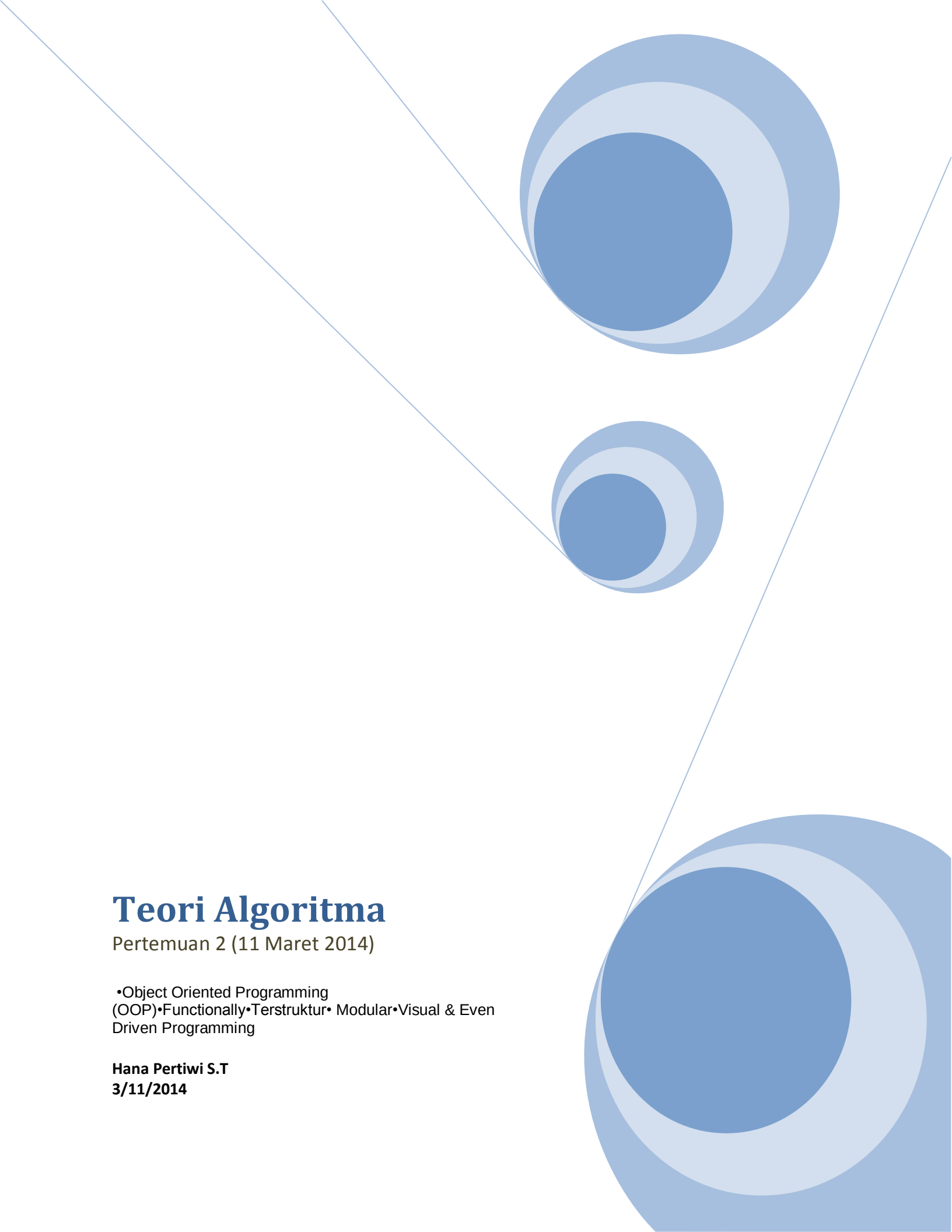


Teori Algoritma

Pertemuan 2 (11 Maret 2014)

•Object Oriented Programming (OOP)•Functionally•Terstruktur• Modular•Visual & Even Driven Programming

Hana Pertiwi S.T

3/11/2014

Object Oriented Programming (OOP)

Suatu metode pemrograman yang berorientasi kepada objek.

Tujuan OOP untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari.

Konsep OOP

1. Encapsulation (enkapsulasi)
2. Abstraction (abstraksi)
3. Inheritance (Pewarisan Sifat)
4. Polymorpishm (Polimorfisme)

Enkapsulasi (encapsulation)

- Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail informasi.
- Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.
- Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.
- Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

Kelas Abstrak (Class Abstraksi)

- Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data.
- Kelas dapat diilustrasikan sebagai suatu cetak biru(blueprint) atau prototipe yang digunakan untuk menciptakan objek.
- Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.
- Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (data field), prosedur atau fungsi (method), dan sifat kelas (property).

Pewarisan (Inheritance)

- Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.
- Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.
- Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.
- Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.

Polimorfisme (polymorphism)

- Polimorfisme merupakan kemampuan objekobjek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.
- Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
- Method overriding.
- Method name overloading.

Pemrograman Prosedural (Procedural Programming) *

- Algoritma berisi urutan langkah-langkah penyelesaian masalah. Ini berarti algoritma adalah proses yang procedural.
- Defenisi procedural adalah :

  1. Tahap-tahap kegiatan untuk menyelesaikan suatu aktivitas
  2. Metode langkah demi langkah secara eksak dalam memecahkan suatu masalah
- Pada pemrograman procedural, program dibedakan antara bagian data dengan bagian instruksi. B*agian instruksi* terdiri atas runtutan instruksi yang dilaksanakan satu persatu secara berurutan oleh pemroses. Alur pelaksanaan instruksi dapat berubah karena adanya pencabangan kondisional. *Data* yang disimpan didalam memori dimanipulasi oleh instruksi secara beruntun atau procedural. Paradigma pemrograman seperti ini dinamakan pemrograman procedural.
- Bahasa-bahasa tingkat tinggi seperti Cobol, Basic, Pascal, Fortran, dan C. mendukung kegiatan pemrograman procedural, karena itu mereka dinamakan juga bahasa procedural.

Pemrograman Terstruktur (Structured Programming) *

- Bahasa pemrograman terstruktur adalah bahasa pemrograman yang mendukung pembuatan program sebagai kumpulan prosedur. Prosedur-prosedur ini dapat saling memanggil dan dipanggil dari manapun dalam program dan dapat menggunakan parameter yang berbeda-beda untuk setiap pemanggilan.
- Prosedur adalah bagian dari program untuk melakukan operasi-operasi yang sudah ditentukan dengan menggunakan parameter tertentu.
- Bahasa pemrograman terstruktur adalah pemrograman yang mendukung abstraksi data, pengkodean terstruktur dan kontrol program terstruktur.
- Kontrol program terstruktur (Tiga tipe Bahasa pemrograman terstruktur):
    1. Terurut (sequence) Setiap baris program akan dikerjakan secara urut dari atas ke bawah (setiap baris dikerjakan sekali atau tidak baris baris program yang tidak dikerjakan)
    2. Pilihan (selection/conditional)
    3. Pengulangan (repetition - loop)

- Prinsip pemrograman terstruktur:

    - ✓ Pendekatan rancangan dari atas ke bawah (top down design),
    - ✓ Bagi program ke dalam modul-modul logika yang sejenis,
    - ✓ Gunakan sub-program untuk proses sejenis yang sering digunakan,
    - ✓ Gunakan pengkodean terstruktur: (IF - THEN, DO-.. WHILE),
    - ✓ Hindarkan penggunaan perintah GO TO bila tidak diperlukan,
    - ✓ Gunakan nama-nama bermakna (mnemonic names), dan
    - ✓ Buat dokumentasi yang akurat dan berarti.
- Gaya penulisan program terstruktur:

    - Menggunakan indentasi sehingga jelas struktur dan kontrol program.
    - Memudahkan pembacaan, pemahaman, penelusuran kesalahan dan pembuatan koreksi.
- Contoh bahasa pemrograman terstruktur : Pascal. Cobol, RPG, ADA, C

Pemrograman Modular(Modular Programming) *

Program-program yang besar cenderung sulit terutama karena kompleksitas dari program tersebut, dan banyak bagian dengan hubungan yang rumit dan detail yang sebenarnya tidak perlu.

Salah satu metode dalam penyusunan program terstruktur adalah pemrograman modular. Dalam pemrograman modular, program dipecah-pecah ke dalam modul-modul, dimana setiap modul

menunjukkan fungsi dan tugas tunggal. Dengan membagi masalah ke dalam modul-modul, maka masalah akan menjadi sederhana sehingga program dapat lebih mudah disusun dan dipahami.

- Fungsi, Prosedur, atau kumpulan perintah-perintah dipaket menjadi suatu modul.
- Dapat digunakan berulang-ulang, atau digunakan olah fungsi, prosedur lain dalam program.

Setiap program mempunyai sebuah modul program utama, yang mengontrol semua proses yang terjadi, termsuk mengirimkan kontrol program ke submodul untuk melakukan suatu fungsi tertentu.

Pemrograman modular diterapkan dengan menggunakan sub-routine, yaitu sebuah kumpulan perintah yang melakukan tugas pemrosesan yang terbatas.

- Jika persoalan yang ingin dipecahkan melalui program terlalu besar, sebaiknya pemecahan masalah dilakukan secara bertahap
- Setiap tahapan akan menghasilkan modul program
- Setiap modul tersebut diberi nama sehingga untuk menyatakannya cukup dengan menyebut namanya
- Deskripsi fungsional dari setiap modul adalah penting

Program yang didefinisikan modulnya dengan baik akan :

- Mudah dibaca dan dimengerti oleh pemakai
- Efisien, karena modul yang sama mungkin dipakai pada beberapa tahapan program.

- Modular programming banyak dimanfaatkan oleh bahasa pemrograman OOP.

Pemrograman Fungsional (Functional Programming) *

- Suatu bahasa dimana ekspresi disusun atas fungsi panggilan (bukan pernyataan). *Fungsi:* perintah-perintah yang terkumpul menjadi satu dan dapat menghasilkan suatu nilai.

- Disebut bahasa pemrograman fungsional karena memang pada program seluruh kodenya berupa fungsi-fungsi. Bahasa pemrograman fungsional merupakan salah satu bahasa pemrograman yang memperlakukan proses komputasi sebagai evaluasi fungsi-fungsi matematika.

- Isi dari Program fungsional tidak mengandung pernyataan perintah.

- Contoh bahasa pemrograman fungsional : Lisp, Scheme, ML, Haskell, Erlang.

Pemrograman Berorientasi Objek* (OOP/Object-Oriented Programming)

Object / Objek : Elemen yang memiliki fungsi, metode, karakteristik tertentu yang dapat dibedakan dalam dunia nyata.

Class : Kumpulan Object-object yang memiliki kesamaan karakateristik.

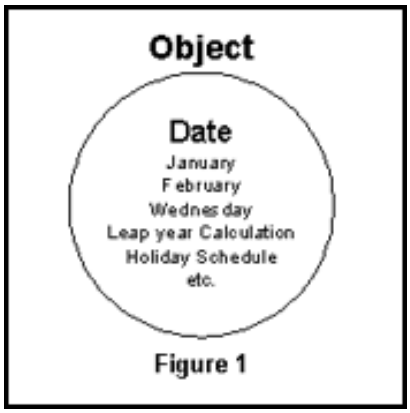


Figure 1

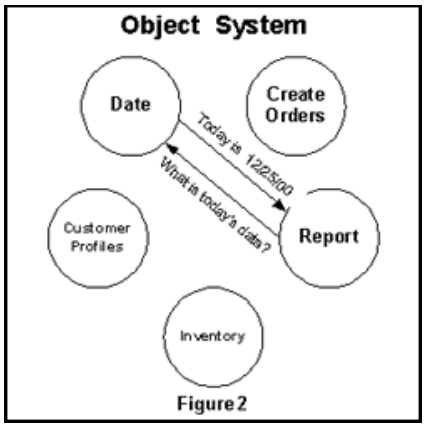


Figure 2

- Merupakan bahasa pemrograman yang mampu memanfaatkan objek-objek yang tersedia atau membuat suatu objek tertentu dengan menggunakan bahasa pemrograman.
- Mampu merefleksikan kebutuhan-kebutuhan user sebagaimana lakyaknya yang ada di dunia nyata.
- Relative lebih fleksibel dan mudah diadaptasikan terhadap perubahan suatu program.
- Memiliki feature yang memperkuat dan meningkatkan flesibilitas suatu objek dengan diadanya class, instance, encapsulation, inheritance, reuseability, dan polymorphism.

- Karakteristik Bahasa Berorientasi Objek:

  - ✓ Objek fisik: (Mobil dalam simulasi arus lalu lintas, Pesawat terbang dalam sistem pengontrolan lalu lintas udara)
  - ✓ Elemen dari lingkungan : (Windows, Objek grafik (garis, lingkaran, polygon))
  - ✓ Penyimpanan data (array, stack, Link list, binary tree)
  - ✓ Entitas orang (karyawan, mahasiswa, pelanggan, pasien)

Contoh bahasa pemrograman beroriantasi object : C++ , SmallTalks , Java

Pemrograman Visual

Bahasa Visual

- Penggunaan ekspresi visual(seperti grafik, gambar, atau ikon) yang sistematik dan mempunyai arti
- Bahasa visual adalah himpunan simbol-simbol grafis dan teks yang mempunyai arti semantik dan digunakan untuk menyelesaikan masalah komunikasi di dunia.

Bandingkan:

Bahasa Textual mengacu pada penggunaan karakter (teks).

Bahasa tekstual konvensional hanya bekerja pada 1 dimensi karena compiler/interpreter memproses program pada satu arah saja.

Pemrograman Visual :

- "Penggunaan ekspresi visual (seperti grafik, gambar, atau ikon) dalam proses pemrograman"

- "Mengacu pada aktivitas yang memungkinkan pengguna untuk membuat program dalam dua (atau lebih) dimensi.

Bahasa Pemrograman Visual:

"Bahasa visual digunakan dalam pemrograman visual"

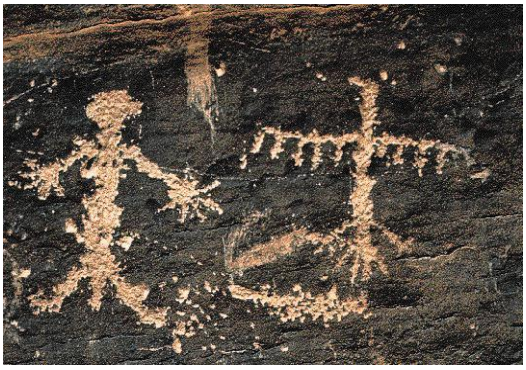
Visualisi:
"Penggunaan representasi visual (grafik, gambar, atau animasi) untuk menggambarkan program, data, struktur atau tingkah laku dinamis sistem yang kompleks."

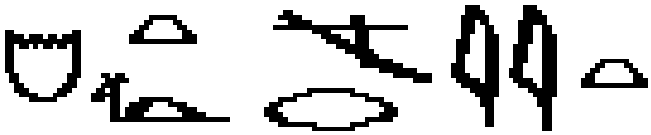
Sistem Pemrograman Visual:

"Sistem komputer yang mendukung baik pemrograman visual maupun visualisasi"

Contoh Bahasa Visual (1)

- λ Kenyataan, bahasa visual
  - lebih tua dari saudaranya tekstual
  - ada anggapan komputer tidak bisa apa-apa
- λ Contoh historis:
  - **Petroglyphs: pahatan pada batu dari masyarakat prasejarah**
  - **Hieroglyphs: Bahsa visual orang Mesir (3000 SM. to 400 M)**





Examples of Visual Languages (2)

- λ Bahasa visual Modern
  - Pictograms
  - Elemen pada GUI
    - Simbol File dan folder, dll.
  - Skema rangkaian elektronik
  - Diagram keadaan, diagram E-R
  - Petri Nets: Bahasa visual untuk tingkah laku system
- λ Bahasa campuran: berisi elemen visual yang mempunyai elemen tekstual (keterangan)

<u>Pemrograman Even-Driven (Even-Driven Programming) *</u>

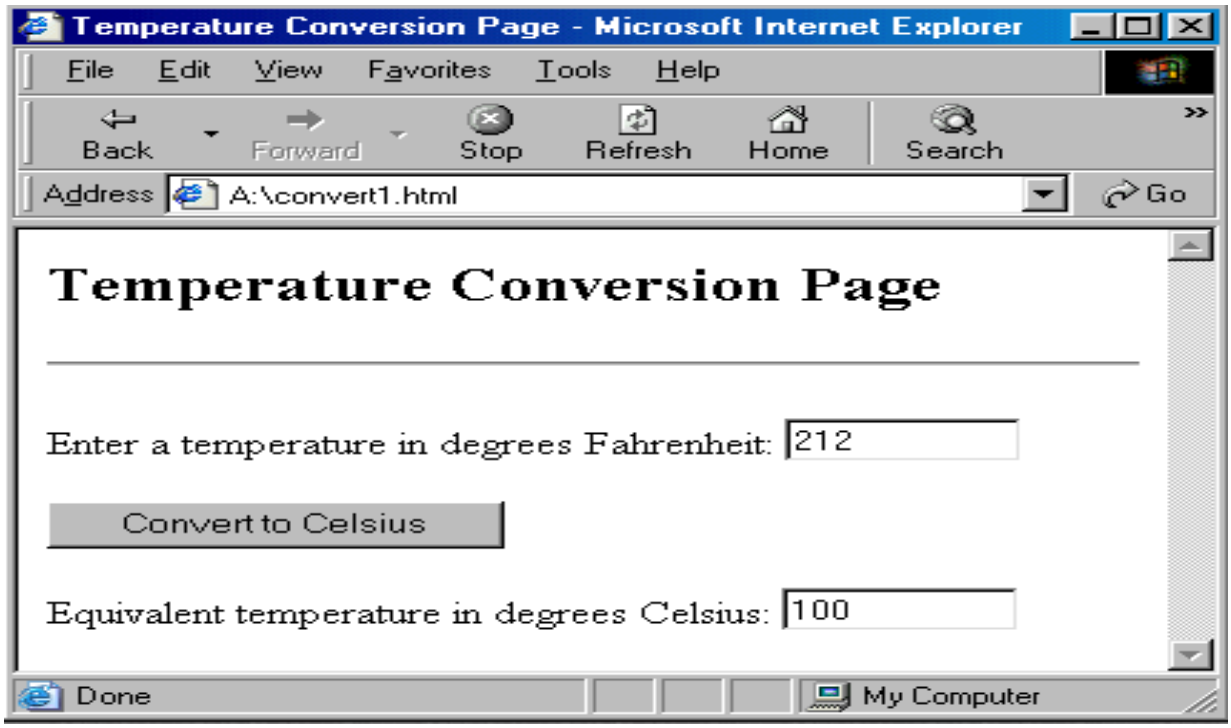
Menggunakan konsep "*Jika sebuah aksi / perintah dilakukan terhadap sebuah objek, apa yang akan terjadi / dilakukan oleh objek tersebut selanjutnya*."

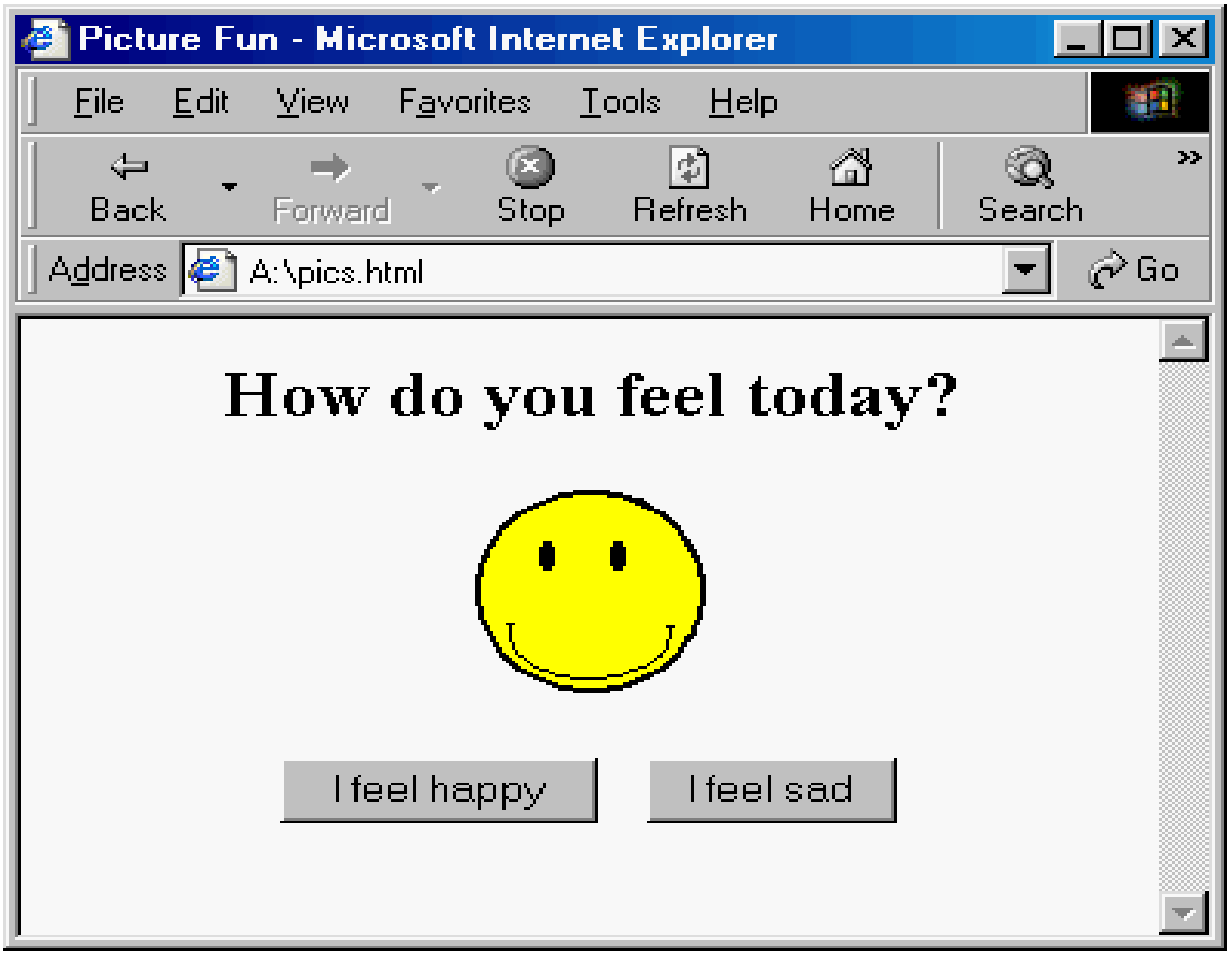
Sangat fleksibel dalam pembuatan koding program, karena sudah menggunakan konsep OOP dimana pemrograman dapat dimulai dari objek yang diinginkan tanpa harus terurut.

Merupakan salah jenis bahasa pemrograman yang sudah memanfaatkan GUI (Graphic User Interface).

Biasanya merupakan jenis bahas pemrograman visual.

Contoh : Visual Basic, Visual C++, Delphi, Borland Kilix


Temperature Conversion Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search
Address A:\convert1.html
Go
Temperature Conversion Page
Enter a temperature in degrees Fahrenheit: 212
Convert to Celsius
Equivalent temperature in degrees Celsius: 100
Done
My Computer



Resensi:

1. http://fatihamaliah.wordpress.com/2013/04/02/pengertian-konsep-oop-object-oriented-programming/
2. **Introduction To Algoritms**, Thomas N. Cormen, Charles E. Leiserson, Ronald L. Ruvest. MIT Press
3. **Computer Algorithms: introduction to design and analysis. 2nd ed.**, Sara Baase, Reading,Mass: Addison-Wesley Company, 1993
4. Analisis dan Desain Berorientasi Objek, Ariesto Hadi Sutopo, JJ Learning: Yogyakarta, 2002

5. Pengantar Analisis Algoritma, Suryadi MT, Gunadarma: Jakarta, 1992
6. Referensi silabus utama:
   http://www.cs.ucl.ac.uk/teaching/syllabus/ug/1b12.htm
   Bisa digunakan: (slides-2)
   http://www.cs.caltech.edu/~cs138/
   http://www.lehigh.edu/~tkr2/teaching/ie170/
   http://hercule.csci.unt.edu/~ian/classes/fall03/csci4450/info.html
   http://highered.mcgrawhill.com/sites/0070131511/student_view0/chapter1/chapter_overview.
7. Neni. "Pertemuan 2". Tersedia: staff.gunadarma.ac.id.